
NLSY97 CODEBOOK SUPPLEMENT

MAIN FILE ROUND 3

**Prepared for the
U.S. Department of Labor by**

**Center for Human Resource Research
The Ohio State University**

**Under contract with
National Opinion Research Center
University of Chicago**

2001

NLSY97 Appendix 3: Family Background and Formation Variable Creation

HOUSEHOLD SIZE AS OF THE SURVEY DATE

Variables Created: CV_HH_SIZE
CV_HH_UNDER_6
CV_HH_UNDER_18

Variables Used

Name in Program	Question Name on CD
hage01-hage14	HHI_AGE.01-.14
huid01-huid14	HHI_UID.01-.14
rsage	SYMBOLKEY!AGE

This program creates several variables describing the composition of the respondent's household: the total number of residents, the number of residents under age 6, and the number of residents under age 18.

```

array hage hage01-hage14;
array huid huid01-huid14;

/* Create dummy variables hhdum[i] (i=1 to 14) that
equal one if the ith member has a member ID, and
zero if the ith member does not have a member ID
(i.e.hhid0i=-4). Also create age dummies: dum6[i]
(i=1 to 14) that equal 1 if the ith household member
is under age 6 and dum18[i] (i=1 to 14) that equal 1 if
the ith household member is under age 18.*/

array hhdum hhdum01-hhdum14;
array dum6 dum601-dum614;
array dum18 dum1801-dum1814;

do i=1 to 14;
    hhdum[i]=0; dum6[i]=0; dum18[i]=0;
end;

do i=1 to 14;
    if huid[i]>-1 then hhdum[i]=1;
    if -1<hage[i]<6 then dum6[i]=1;
    if -1<hage[i]<18 then dum18[i]=1;
end;

/* Create the dummy for the respondent's age rdum6
and rdum18. */
rdum6=0; rdum18=0;

if -1<rsage<6 then rdum6=1;
if -1<rsage<18 then rdum18=1;

```

```

/* Create household size hhsiz by adding up the
dummies hhdum[i] and also add one due to the
respondent. Similarly, create variabes under6 and
under18 by adding up the other two dummies.*/

hhsiz=1; under6=rdum6; under18=rdum18;

do i=1 to 14;
    hhsiz=hhsiz+hhdum[i];
    under6=under6+dum6[i];
    under18=under18+dum18[i];
end;

if huid01=-5 then hhsiz=-5;
if -4<huid01<0 or -4<huid02<0 or -4<huid03<0 or
-4<huid04<0 or -4<huid05<0 or -4<huid06<0 or
-4<huid07<0 or -4<huid08<0 or -4<huid09<0 or
-4<huid10<0 or -4<huid11<0 or -4<huid12<0 or
-4<huid13<0 or -4<huid14<0 then hhsiz=-3;

if huid01=-5 then under6=-5;
if huid01=-5 then under18=-5;
if -4<hage01<0 or -4<hage02<0 or -4<hage03<0 or
-4<hage04<0 or -4<hage05<0 or -4<hage06<0 or
-4<hage07<0 or -4<hage08<0 or -4<hage09<0 or
-4<hage10<0 or -4<hage11<0 or -4<hage12<0 or
-4<hage13<0 or -4<hage14<0 or -4<rsage<0 then
do;
    under6=-3; under18=-3;
end;

endsas;

```

YOUTH'S RELATIONSHIP TO HOUSEHOLD PARENT FIGURE(S)

Variables Created: CV_YTH_REL_HH_CURRENT

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
Round 1:		Round 3:	
pubid	PUBID	hhage01-hhage14	HHI_AGE.01-.14
yid	YOUTH_ID.01	hh2uid01-hh2uid14	HHI_UID.01-.14
sh931-sh935	SH-93.01-.05	marry_1-marry_14	HHI_MARSTAT.01-.14
hh1uid01-hh1uid16	HHI2_UID.01-.16	rel_1-rel_14	HHI_RELY.01-.14
		gend_1-gend_14	HHI_SEX.01-.14
		age	CV AGE INT DATE 1999

Codes for Created Variable

- | | |
|------------------------------------|---------------------------------|
| 1 = Both biological parents | 6 = Adoptive parent(s) |
| 2 = Two parents, biological mother | 7 = Foster parent(s) |
| 3 = Two parents, biological father | 8 = No parents, grandparents |
| 4 = Biological mother only | 9 = No parents, other relatives |
| 5 = Biological father only | 10 = Anything else |

This program creates a variable identifying the youth's relationship to the primary adults in the household. In round 3 (as in round 2) there was no information collected on the legal guardian of the youth, so it is not possible to determine whether respondents are living with non-parent relatives because they are guardians or because the living situation is better (e.g., closer to school, no rent). For this reason, youths above the age 18 and above are considered independent and, if they are not living with an identified parent or parent-figure (legal guardian), are put into the anything else category. Youths below the age of 18 who are not living with an identified parent or parent-figure are put into the category that most closely matches their household situation.

```

array rel_a (i) rel_1-rel_14;           array marry_a (i) marry_1-marry_14;
array gend_a (i) gend_1-gend_14;   array age_a (i) hhage01-hhage14;

/* this part determines legal guardians so that we can determine whether any guardians are present from the R1
interview use only unique IDs (cases with 6 digits are hh members after R1 only, so this program won't do anything
with them, as we don't have info on legal guardian status for these HH members) */

array hh1uid (i) hh1uid01-hh1uid14;      *round 1 variable;
array hh2uid (i) hh2uid01-hh2uid14;      *round 3 variable;

do i=1 to 14;
  if yid=1 and sh931>0 and sh931+100=hh1uid then do; uid=hh1uid; end;
  if yid=2 and sh932>0 and sh932+100=hh1uid then do; uid=hh1uid; end;
  if yid=3 and sh933>0 and sh933+100=hh1uid then do; uid=hh1uid; end;
  if yid=4 and sh934>0 and sh934+100=hh1uid then do; uid=hh1uid; end;
  if yid=5 and sh935>0 and sh935+100=hh1uid then do; uid=hh1uid; end;
end;

do i=1 to 14; if (uid ne . and uid=hh2uid) then line=i; end;

legal=0;
do i=1 to 14; if line=i then legal=rel_a; end;

momid=0;      domid=0;      adopdad=0;      admom=0;      fostma=0;      fostda=0;
stepma=0;      stepda=0;      husb=0;          wife=0;          grand=0;          relat=0;
nonrel=0;      indep=0;      spouse=0;

```

```

do i=1 to 14;
  if (legal>28 and legal<37) then do; grand=1; end;
  if legal=1 or legal=2 then do; spouse=1; end;    *spouse;
  if (legal>12 and legal<19) and age_a>20 then do; relat=1; end;    *brother/sister;
  if (legal>69 and legal<85) and age_a>20 then do; relat=1; end;    *aunt or uncle and other relatives;
  *lover, roommate, other non-relative, mom's or dad's partner;
  if legal=69 or legal=68 or legal=85 or legal=89 then do; nonrel=1; end;
  if legal=-1 or legal=-2 or legal=-3 then do; invalid=1; end;    *missing values;
  if rel_1=-4 and rel_2=-4 and rel_3=-4 and rel_4=-4 and rel_5=-4 and rel_6=-4 and rel_7=-4 and
      rel_8=-4 and rel_9=-4 and rel_10=-4 and rel_11=-4 and rel_12=-4 and rel_13=-4 and rel_14=-4 then do;
      indep=1;
  end;
end;
**not legal;
do i=1 to 14;
  if (rel_a>28 and rel_a<37) then do; nlgrand=1; end;
  if rel_a=1 or rel_a=2 then do; nlpouse=1; end;    *spouse;
  if (rel_a>12 and rel_a<19) and age_a>20 then do; nlrelat=1; end;    *brother/sister;
  if (rel_a>69 and rel_a<85) and age_a>20 then do; nlrelat=1; end;    *aunt or uncle and other relatives;
  *lover, roommate, other non-relative, mom's or dad's partner;
  if rel_a=69 or rel_a=68 or rel_a=85 or rel_a=88 or rel_a=89 then do; nlnrnl=1; end;
  if rel_a=-1 or rel_a=-2 or rel_a=-3 then do; nlinv=1; end;
end;

if age<18 then do;
  if nlgrand ne . and legal=0 then grand=nlgrand;
  if nlpouse ne . and legal=0 then spouse=nlpouse;
  if nlrelat ne . and legal=0 then relat=nlrelat;
  if nlnrnl ne . and legal=0 then nonrel=nlnrnl;
  if nlinv ne . and legal=0 then invalid=nlinv;
end;

*for all youths;
if nlpouse ne . then spouse=nlpouse;

do i=1 to 14;
  if rel_a=3 then momid=i;          if rel_a=4 then domid=i;
  if rel_a=5 then admom=i;         if rel_a=6 then adopdad=i;
  if rel_a=7 then stepma=i;        if rel_a=8 then stepda=i;
  if rel_a=9 then fostma=i;        if rel_a=10 then fostda=i;
end;
rel=-16;
if age>17 then do; rel=10; end;           if marry_1=-5 then do; rel=-5; end;
if indep=1 then do; rel=10; end;           if invalid=1 then do; rel=-3; end;
if nonrel>0 then do; rel=10; end;          if relat>0 then do; rel=9; end;
if grand>0 and momid=0 and domid=0 then do; rel=8; end;
if spouse=1 then do; rel=10; end;          if fostda>0 or fostma>0 then do; rel=7; end;
if admom>0 or adopdad>0 then do; rel=6; end; if stepda>0 or stepma>0 then do; rel=10; end;
if momid>0 and domid=0 then do; rel=5; end; if momid>0 and domid=0 then do; rel=4; end;

if domid>0 and momid=0 then do; if admom>0 or stepma>0 then rel=3; end;
if momid>0 and domid=0 then do; if adopdad>0 or stepda>0 then rel=2; end;
if momid>0 and domid>0 then do; both=1; rel=1; end;
if rel=-16 then rel=10;

```

endsas;

HIGHEST GRADE COMPLETED BY YOUTH'S PARENTS

Variables Created:	CV_HGC_BIO_DAD	CV_HGC_RES_DAD
	CV_HGC_BIO_MOM	CV_HGC_RES_MOM

Variables Used

Name in Program	Round 1 Question Name on CD	Name in Program	Round 1 Question Name on CD
hhid	YOUTH_ID.01	hhrel121-hhrel125	HHI2_REL12.01-05
pubid	PUBID	hhrel131-hhrel135	HHI2_REL13.01-05
hhg1-hhg16	HHI2_HIGHGRADE.01-.16	hhrel141-hhrel145	HHI2_REL14.01-05
hhrel11-hhrel15	HHI2_REL1.01-.05	hhrel151-hhrel155	HHI2_REL15.01-05
hhrel21-hhrel25	HHI2_REL2.01-.05	hhrel161-hhrel165	HHI2_REL16.01-05
hhrel31-hhrel35	HHI2_REL3.01-.05	nhg1-nhg21	NONHHI_HIGHGRADE.01-.21
hhrel41-hhrel45	HHI2_REL4.01-.05	nhrel1-nhrel21	NONHHI_RELATION.01-.21
hhrel51-hhrel55	HHI2_REL5.01-.05	chkhgc	PINF-085A
hhrel61-hhrel65	HHI2_REL6.01-.05	crcthgc	PINF-085B
hhrel71-hhrel75	HHI2_REL7.01-.05	resparid	YOUTH_PARENTID.01
hhrel81-hhrel85	HHI2_REL8.01-.05	ysaq056	YSAQ-056
hhrel91-hhrel95	HHI2_REL9.01-.05	ysaq057	YSAQ-057
hhrel101-hhrel105	HHI2_REL10.01-.05	ysaq145	YSAQ-145
hhrel111-hhrel115	HHI2_REL11.01-.05	ysaq146	YSAQ-146

This program uses the household and nonresident relative rosters from the round 1 survey to create variables for the education level of respondents' biological parents and resident parent-figures. Users should note that if the respondent's biological parents are also the resident parent-figures, they will be listed in both variables. Additionally, these variables are based on round 1 data but are included in this codebook supplement because they were first created and released with the round 3 data.

```

if ysaq057=1 then ysaq057=0;      if ysaq057=2 then ysaq057=1;      if ysaq057<0 then ysaq057=0;
if ysaq146=1 then ysaq146=0;      if ysaq146=2 then ysaq146=1;      if ysaq146<0 then ysaq146=0;

chkhgc=pinf085A;      crcthgc=pinf085B;

ncmom=0;                  ncdad=0;
if ysaq056=0 then ncmom=1;    if ysaq145=0 then ncdad=1;

array hhrel1 hhrel11 hhrel21 hhrel31 hhrel41 hhrel51 hhrel61 hhrel71 hhrel81 hhrel91 hhrel101 hhrel111 hhrel121
      hhrel131 hhrel141 hhrel151 hhrel161;
array hhrel2 hhrel12 hhrel22 hhrel32 hhrel42 hhrel52 hhrel62 hhrel72 hhrel82 hhrel92 hhrel102 hhrel112 hhrel122
      hhrel132 hhrel142 hhrel152 hhrel162;
array hhrel3 hhrel13 hhrel23 hhrel33 hhrel43 hhrel53 hhrel63 hhrel73 hhrel83 hhrel93 hhrel103 hhrel113 hhrel123
      hhrel133 hhrel143 hhrel153 hhrel163;
array hhrel4 hhrel14 hhrel24 hhrel34 hhrel44 hhrel54 hhrel64 hhrel74 hhrel84 hhrel94 hhrel101 hhrel114 hhrel124
      hhrel134 hhrel144 hhrel154 hhrel164;
array hhrel5 hhrel15 hhrel25 hhrel35 hhrel45 hhrel55 hhrel65 hhrel75 hhrel85 hhrel95 hhrel101 hhrel115 hhrel125
      hhrel135 hhrel145 hhrel155 hhrel165;
array hhg hhg1-hhg16;
array nhg nhg1-nhg21;
array nhrel nhrel1-nhrel21;

/* find the line number of the respondents' resident parents and the education level*/
hhmomed=-4;   hhdaded=-4;   bmomed=-4;   bdaded=-4;
bmomed1=-4;   bdaded1=-4;   hhmom=0;     hhdad=0;

```

Appendix 3: Family Background and Formation Variable Creation

```
/*the following vars indicate cases where Responding parent differed from screener parent and R parent changed  
the HGC information collected in the initial screener. Parent's HGC variables created below use this corrected  
data*/  
flagmom=0;      flagdad=0;  
  
if hhid=1 then do;  
do i=1 to 16;  
  if hhrel1[i] in (3,5,7,9) then do; hhmomed=hhg[i]; if hhg[i]=-4 then hhmomed=-3; end;  
    if hhrel1[i] in (4,6,8,10) then do; hhdaded=hhg[i]; if hhg[i]=-4 then hhdaded=-3; end;  
    if hhrel1[i]=3 then do; bmomed=hhg[i]; if hhg[i]=-4 then bmomed=-3; end;  
    if hhrel1[i]=4 then do; bdaded=hhg[i]; if hhg[i]=-4 then bmomed=-3; end;  
  end;  
  if hhrel11 in (99,-1,-2,-3) or hhrel21 in (99,-1,-2,-3) or hhrel31 in (99,-1,-2,-3) or hhrel41 in (99,-1,-2,-3) or  
    hhrel51 in (99,-1,-2,-3) or hhrel61 in (99,-1,-2,-3) or hhrel71 in (99,-1,-2,-3) or hhrel81 in (99,-1,-2,-3) or  
    hhrel91 in (99,-1,-2,-3) or hhrel101 in (99,-1,-2,-3) or hhrel111 in (99,-1,-2,-3) or hhrel121 in (99,-1,-2,-3) or  
    hhrel131 in (99,-1,-2,-3) or hhrel141 in (99,-1,-2,-3) or hhrel151 in (99,-1,-2,-3) or hhrel161 in (99,-1,-2,-3)  
  then do;  
    if hhmomed=-4 then hhmomed=-3;      if hhdaded=-4 then hhdaded=-3;  
    if bmomed=-4 then bmomed=-3;          if bdaded=-4 then bdaded=-3;  
  end;  
/*this part of the loop uses screener information (from PINF) on corrections to HGC of responding parent*/  
do i=1 to 16;  
  if resparid=i and hhrel1[i] in (3,5,7,9) and chkhgc=1 then do;  
    flagmom=1; hhmomed=crcthgc;  
    if hhrel1[i]=3 then bmomed=crcthgc;  
  end;  
  if resparid=i and hhrel1[i] in (4,6,8,10) and chkhgc=1 then do;  
    flagdad=1; hhdaded=crcthgc;  
    if hhrel1[i]=4 then bdaded=crcthgc;  
  end;  
end;  
end;
```

/* At this point the program repeats the same code for hhid=2, hhid=3, hhid=4, and hhid=5. This code is not included due to space considerations. Researchers needing more information should contact NLS User Services.

```
/* if birth parents are not in the household */  
/*the two following dummies flag cases where a Bio parent is indicated on the non-household roster, but HGC reported  
for this parent=-4. The HGC value in the created variable is assigned as -3, as these are not valid skips. */  
nhmom=0;      nhdad=0;  
  
do i=1 to 21;  
  if nhrel[i]=3 then do;  
    bmomed1=nhg[i];  
    if nhg[i]=-4 then do; bmomed1=-3; nhmom=1; end;  
  end;  
  if nhrel[i]=4 then do;  
    bdaded1=nhg[i];  
    if nhg[i]=-4 then do; bdaded1=-3; nhdad=1; end;  
  end;  
end;  
  
/* check if there is some conflict between the two answers about the birth parents. */  
flagm=0;      flagd=0;  
if bmomed not in (-4,-3) and bmomed1 ne -4 and bmomed ne bmomed1 then flagm=1;
```

Appendix 3: Family Background and Formation Variable Creation

```
if bdaded not in (-4,-3) and bdaded1 ne -4 and bdaded ne bdaded1 then flagd=1;

/* Get the final data about the birth parents' education. */
if bmomed in (-4,-3) and bmomed1 ne -4 then bmomed=bmomed1;
if bdaded in (-4,-3) and bdaded1 ne -4 then bdaded=bdaded1;

/* hand edit for cases with conflicts between household and nonhousehold roster about the birth mom and birth
dad's education. Cases 1-3 were determined by inspection. Case 4 was determined from the rounds 2 and 3
household and nonhousehold rosters */

if pubid=3689 then bdaded=13;      if pubid=4210 then bdaded=10;
if pubid=5457 then bdaded=4;       if pubid=6579 then bmomed=8;

/* hand edit for one person. This case is from the same household as 0018, so use that person's information */
if pubid=0019 then bdaded=10;

/*recode as missing cases where there was no HH screener*/
if pubid=7045 then do; bdaded=-3; bmomed=-3; hhmomed=-3; hhdaded=-3; end;
if pubid=8767 then do; bdaded=-3; bmomed=-3; hhmomed=-3; hhdaded=-3; end;

/*recode all missing data to -3*/
if bmomed in (-1,-2,-3) then bmomed=-3;           if bdaded in (-1,-2,-3) then bdaded=-3;
if hhmomed in (-1,-2,-3) then hhmomed=-3;         if hhdaded in (-1,-2,-3) then hhdaded=-3;

/*Recode parents with a value of "0" for HGC as MISSING (due to apparent problems with HH and NonHH roster
information--it is unclear whether these individuals actually NEVER attended any school, or if there was an error in
the rosters. Also, create a flag for these cases (there were 82 cases where one or more of the parent's education
variables=0) so if roster data are corrected these cases can be easily fixed*/

array educ{*} hhmomed hhdaded bdaded bmomed;
zeroflag=0;
do i=1 to 4;
  if educ{i}=0 then do; educ{i}=-3; zeroflag=1; end;
end;
endsas;
```

YOUTH'S MARITAL STATUS AND MARITAL/COHABITATION HISTORY

Variables Created:	CV_MARSTAT CV_FIRST_COHAB_DATE_M CV_FIRST_MARRY_DATE_M CV_FIRST_COHAB_MONTH CV_COHAB_TTL	CV_MARSTAT_COLLAPSED CV_FIRST_COHAB_DATE_Y CV_FIRST_MARRY_DATE_Y CV_FIRST_COHAB_MONTH CV_MARRIAGES_TTL
---------------------------	--	--

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
Round 1 variables:			
fcoh_mr1	CV_FIRST_COHAB_DATE_M	int_mr1	CV_INTERVIEW_DATE_M
fcoh_yr1	CV_FIRST_COHAB_DATE_Y	int_yr1	CV_INTERVIEW_DATE_Y
fcohcmr1	CV_FIRST_COHAB_MONTH	marstat1	CV_MARSTAT
fmar_mr1	CV_FIRST_MARRY_DATE_M	cmarsta1	CV_MARSTAT_COLLAPSED
fmar_yr1	CV_FIRST_MARRY_DATE_Y	coh_ttl1	CV_COHAB_TTL
fmarmr1	CV_FIRST_MARRY_MONTH	mar_ttl1	CV_MARRIAGES_TTL
int_dr1	CV_INTERVIEW_DATE_D	pubid	PUBID
Round 2 variables:			
m45001r2	YMAR-4500.01	fmarmr2	CV_FIRST_MARRY_MONTH
m45002r2	YMAR-4500.02	int_dr2	CV_INTERVIEW_DATE~D
m54011r2	YMAR-5400.01.01	int_mr2	CV_INTERVIEW_DATE~M
m54012r2	YMAR-5400.01.02	int_yr2	CV_INTERVIEW_DATE~Y
fcoh_mr2	CV_FIRST_COHAB_DATE~M	marstat2	CV_MARSTAT
fcoh_yr2	CV_FIRST_COHAB_DATE~Y	cmarsta2	CV_MARSTAT_COLLAPSED
fcohcmr2	CV_FIRST_COHAB_MONTH	coh_ttl2	CV_COHAB_TTL
fmar_mr2	CV_FIRST_MARRY_DATE~M	mar_ttl2	CV_MARRIAGES_TTL
fmar_yr2	CV_FIRST_MARRY_DATE~Y		
Round 3 variables:			
m620	YMAR-620	m540011, m540012	YMAR-5400.01.01, .02
m650	YMAR-650	m560011m, y	YMAR-5600.01.01~M, ~Y
m700	YMAR-700	m560012m, y	YMAR-5600.01.02~M, ~Y
m710	YMAR-710	m570011m, y	YMAR-5700.01.01~M, ~Y
m712	YMAR-712	m63001, m63002	YMAR-6300.01, .02
m714	YMAR-714	m700011m, y	YMAR-7000.01.01~M, ~Y
m718	YMAR-718	m700012m, y	YMAR-7000.01.02~M, ~Y
m729d	YMAR-729D	m700013m, y	YMAR-7000.01.03~M, ~Y
m730	YMAR-730	m700021m, y	YMAR-7000.02.01~M, ~Y
m740	YMAR-740	m727021	YMAR-7270.01.01, .02.01
m760	YMAR-760	m730011, m730021	YMAR-7300.01.01, .02.01
m1000	YMAR-1000	m7900111, m7900112	YMAR-7900.01.01.01, .02.01.01
m1500	YMAR-1500	m810111m, y	YMAR-8100.01.01.01~M, ~Y
m30501, m30502	YMAR-3050.01, .02	m820111m, y	YMAR-8200.01.01.01~M, ~Y
m31001m, y	YMAR-3100.01~M, ~Y	m820211m, y	YMAR-8200.02.01.01~M, ~Y
m31002m, y	YMAR-3100.02~M, ~Y	m920011m, y	YMAR-9200.01.01~M, ~Y
m45001, m45002	YMAR-4500.01, .02	m920012m, y	YMAR-9200.01.02~M, ~Y
m46501, m46502	YMAR-4650.01, .02	m920021m, y	YMAR-9200.02.01~M, ~Y
m46701	YMAR-4670.01	m9800111	YMAR-9800.01.01.01
m47001, m47002	YMAR-4700.01, .02	m114001, m114002	YMAR-11400.01, .02
m48001m, y	YMAR-4800.01~M, ~Y	int_d, m, y	CV_INTERVIEW_DATE~D, ~M, ~Y
m48002m, y	YMAR-4800.02~M, ~Y		

Codes for Created Variable

Marital/Cohabitation Status

- | | |
|-----------------------------------|-------------------------------|
| 1 = never married, cohabiting | 6 = separated, not cohabiting |
| 2 = never married, not cohabiting | 7 = divorced, cohabiting |
| 3 = married, spouse present | 8 = divorced, not cohabiting |
| 4 = married, spouse absent | 9 = widowed, cohabiting |

5 = separated, cohabiting

10 = widowed, not cohabiting

Collapsed Marital Status

0 = never married

1 = married

2 = separated

3 = divorced

4 = widowed

This program creates two variables that describe marital status/cohabitation status as of the interview date for respondents age 16 and older. Other respondents are valid skips (-4). Note that later partners take precedence over earlier partners.

The program also creates variables that provide the dates of the youth's first marriage and/or cohabitation in both a continuous month scheme and as actual dates (for more information on the continuous month scheme, see appendix 7 in this document). Summary variables count the total number of marriages and cohabitutions for each youth. Note that these variables are available only for youths age 16 and older as of 12/31/98. If a respondent is cohabiting and then marries it is considered both a cohabitation and a marriage. If someone refuses or doesn't know the full date of their marriage or cohabitation, then the spell is counted in the total variables and the date variables are coded -1 or -2 as applicable.

```
/**Create flag indicating whether r. said there was a
spouse in the house in the hh roster**/

spo_hous=0;
array hhrel (i) hhrel01-HHREL14;

do i=1 to 14;
  if hhrel=1 or hhrel=2 then spo_hous=1;
end;

/*Create a variable indicating r.'s marital status as they
report it*/
if m650 ge 0 then marques=m650;
else marques=m620;

/* Create a variable with r.'s previous marital status*/
prevmsta=MARSTAT2;  prevcmst=CMARSTA2;
mar_ttl=mar_ttl2;    coh_ttl=coh_ttl2;
prevmary=FMAR_Yr2;   prevmarm=FMAR_Mr2;
prvmarcm=fmarmr2;   prevcohm=FCOH_Mr2;
prevcohy=FCOH_Yr2;   prvcohcm=FCOHCMr2;

if cmarsta2=-5 then do;
  prevmsta=marstat1;  prevcmst=cmarsta1;
  mar_ttl=mar_ttl1;    coh_ttl=coh_ttl1;
  prevmary=FMAR_Yr1;   prevmarm=FMAR_Mr1;
  prvmarcm=fmarmr1;   prevcohm=FCOH_Mr1;
  prevcohy=FCOH_Yr1;   prvcohcm=FCOHCMr1;
end;

/*Hand edit previous marital status based on errors*/
if pubid=7891 then do;
  prevmsta=2; prevcmst=0;
end;
```

```
/*Create interview date and previous interview date in
continuous months and months/years*/
if int_y gt 0 and int_m gt 0 then do;
  iym=int_y*100+int_m;
  doicm=(int_y-1980)*12+int_m;
end;
else if int_y eq -5 or int_m eq -5 then do;
  iym=-5;
  doicm=-5;
end;
else if int_y in(-1,-2, -3,-4) or int_m in(-1, -2, -3, -4)
then do;
  iym=-3;
  doicm=-3;
end;

if int_yr2 gt 0 and int_mr2 gt 0 then do;
  dliym=int_yr2*100+int_mr2;
  dlicm=(int_yr2-1980)*12+int_mr2;
end;
else if int_yr2 = -5 then do;
  dliym=int_yr1*100+int_mr1;
  dlicm=(int_yr1-1980)*12+int_mr1;
end;
else if int_yr2 in(-1,-2, -3,-4) or int_mr2 in(-1, -2, -3, -4)
then do;
  dliym=-3;
  dlicm=-3;
end;

/*Create birthdate in continuous months*/
b14cm=(bdate_y+14-1980)*12+bdate_m;

/*set up arrays with marital/cohab status for each of the
continuous months*/
```

<pre> array m (l) m001-m244; /* married y/n */ array coh (l) coha001-coha244; /* cohab y/n */ array mars (l) mars001-mars244; /* marital status */ /*initialize values to 0 for all who go through section*/ if m700>-4 then do; marstat=2; ttlm=0; ttlc=0; do l=1 to 244; if dlicm le L le doicm then do; m=0; coh=0; mars=2; end; end; /*year start cohabitating*/ array ysca (t) M31001Y M700011Y; array yscb (t) M31002Y yscb2; array yscc (t) M31003Y yscc2; /*month start cohabitating*/ array msca (t) M31001M M700011M; array mscb (t) M31002M mscb2; array mscc (t) M31003M mscc2; /*collapse all start years and months into arrays*/ array ysc (p) ysca yscb yscc; array msc (p) msca mscb mscc; /*legally married when began cohabitating*/ array mbega (t) M45001 mbega2; array mbegb (t) M45002 mbegb2; array mbegc (t) M45003 mbegc2; /*collapse married into array*/ array mbeg (p) mbega mbegb mbegc; /*dummy for cohabitation continuous - since dli*/ array cdlia (t) M46701 cdlia2; array cdlib (t) M46702 cdlib2; array cdlic (t) cdlic1 cdlic2; /*dummy for cohabitation continuous - since start (after dli date)*/ array cbega (t) M47001 cbega2; array cbegb (t) M47002 cbegb2; array cbeg2a (t) M910011 cbeg2a2; array cbeg2b (t) cbeg2b1 cbeg2b2; array cbeg2c (t) cbeg2c1 cbeg2c2; /*collapse continuous cohab vars into array*/ array cdli (p) cdlia cdlib cdlic; array cbeg (p) cbega cbegb cbegc; array cbeg2 (p) cbeg2a cbeg2b cbeg2c; /*year end cohabitating (not continuous)*/ array yeca (t) M48001Y M920011Y; array yecb (t) M48002Y yecb2; </pre>	<pre> array yecc (t) M48003Y yecc2; /*month end cohabitating (not continuous)*/ array meca (t) M48001M M920011M; array mecb (t) M48002M mecb2; array mecc (t) M48003M mecc2; /*collapse all end years and months into arrays*/ array yec (p) yeca yecb yecc; array mec (p) meca mecb mecc; /*first or next change in status*/ array howa1 (t) M540011 M9800111; array howa2 (t) M540012 howa22; array howa3 (t) M7900111 howa32; array howb3 (t) M7900211 howb32; array howa4 (t) M7900112 howa42; array howa5 (t) M7900113 howa52; /*collapse into how status changed*/ array how1 (p) howa1 howb1 howc1; array how2 (p) howa2 howb2 howc2; array how3 (p) howa3 howb3 howc3; array how4 (p) howa4 howb4 howc4; array how5 (p) howa5 howb5 howc5; /*month and year change is marry*/ array hyma1 (t) M570011Y M101111Y; array hmma1 (t) M570011M M101111M; array hyma2 (t) M570012Y hyma22; array hmma2 (t) M570012M hmma22; array hyma3 (t) M820111Y hyma32; array hymb3 (t) M820211Y hymb32; array hmma3 (t) M820111M hmma32; array hmmb3 (t) M820211M hmmb32; /*collapsing into month and year change is marry*/ array hym1 (p) hyma1 hymb1 hymc1; array hmm1 (p) hmma1 hmmb1 hmmc1; array hym2 (p) hyma2 hymb2 hymc2; array hmm2 (p) hmma2 hmmb2 hmmc2; array hym3 (p) hyma3 hymb3 hymc3; array hmm3 (p) hmma3 hmmb3 hmmc3; array hym4 (p) hyma4 hymb4 hymc4; array hmm4 (p) hmma4 hmmb4 hmmc4; /*month & year change=separation, divorce, annulment*/ array hysa2 (t) M560012Y hysa22; array hmsa2 (t) M560012M hmsa22; array hysa3 (t) M810111Y hysa32; array hmsa3 (t) M810111M hmsa32; array hysa4 (t) M810112Y hysa42; array hmsa4 (t) M810112M hmsa42; </pre>
---	---

```

/*collapse into month and year change=separation,
   divorce, annulment*/
array hys1 (p) hysa1 hysb1 hysc1;
array hms1 (p) hmsa1 hmsb1 hmsc1;
array hys2 (p) hysa2 hysb2 hysc2;
array hms2 (p) hmsa2 hmsb2 hmsc2;
array hys3 (p) hysa3 hysb3 hysc3;
array hms3 (p) hmsa3 hmsb3 hmsc3;
array hys4 (p) hysa4 hysb4 hysc4;
array hms4 (p) hmsa4 hmsb4 hmsc4;

/*year and month start cohabitating*/
array ymsca (t) ymsca1 ymsca2;
array ymscb (t) ymscb1 ymscb2;
array ymscc (t) ymscc1 ymscc2;
   array ymsc (p) ymsca ymscb ymscc;

/*year and month end cohabitating*/
array ymeca (t) ymeca1 ymeca2;
array ymecb (t) ymecb1 ymecb2;
array ymecc (t) ymecc1 ymecc2;
   array ymec (p) ymeca ymecb ymecc;

/*continuous start month of cohabitation*/
array csmca (t) csmca1 csmca2;
array csmcb (t) csmcb1 csmcb2;
array csmcc (t) csmcc1 csmcc2;
   array csmc (p) csmca csmcb csmcc;

/*continuous end month of cohabitation*/
array cemca (t) cemca1 cemca2;
array cemcb (t) cemcb1 cemcb2;
array cemcc (t) cemcc1 cemcc2;
   array cemc (p) cemca cemcb cemcc;

/*year and month start marriage*/
array ymsma (t) ymsma1-ymsma2;
array ymsmb (t) ymsmb1-ymsmb2;
array ymsmc (t) ymsmc1-ymsmc2;
   array ymsm (p) ymsma ymsmb ymsmc;

/*continuous start month of marriage*/
array csmma (t) csmma1-csmma2;
array csmbb (t) csmbb1-csmbb2;
array csmmc (t) csmmc1-csmmc2;
   array csmm (p) csmma csmbb csmmc;

/*month and year of start separation, etc*/
array ymssa (t) ymssa1-ymssa2;
array ymssb (t) ymssb1-ymssb2;
array ymssc (t) ymssc1-ymssc2;
   array ymss (p) ymssa ymssb ymssc;

/*continuous start month of separation, etc*/
array csmsa (t) csmsa1-csmsa2;
array csmsb (t) csmsb1-csmsb2;

```

```

array csmsc (t) csmsc1-csmsc2;
   array csms (p) csmsa csmsb csmsc;

/*married at date of last interview*/
array mdlia (t) prevcmst mdlia2;
array mdlib (t) mdlib1 mdlib2;
array mdlic (t) mdlic1 mdlic2;
   array mdli (p) mdlia mdlib mdlic;

/*Collapsed marital status*/
array cmarsa (t) cmarsa1 cmarsa2;
array cmarsb (t) cmarsb1 cmarsb2;
array cmarsc (t) cmarsc1 cmarsc2;
   array cmars (p) cmarsa cmarsb cmarsc;

/*number of cohabitations*/
array nuca (t) nuca1 nuca2;
array nueb (t) nueb1 nueb2;
array nucc (t) nucc1 nucc2;
   array nuc (p) nuca nueb nucc;

/*number of marriages*/
array numa (t) numal numa2;
array numb (t) numb1 numb2;
array numc (t) numc1 numc2;
   array num (p) numa numb numc;

/*dummy set to one if start month cohabitare is
invalid*/
array fixmc1 (t) fixmc11 fixmc12;
array fixmc2 (t) fixmc21 fixmc22;
array fixmc3 (t) fixmc31 fixmc32;
   array fixmc (p) fixmc1 fixmc2 fixmc3;

/*dummy set to one if start month marry is invalid*/
array fixmm1 (t) fixmm11 fixmm12;
array fixmm2 (t) fixmm21 fixmm22;
array fixmm3 (t) fixmm31 fixmm32;
   array fixmm (p) fixmm1 fixmm2 fixmm3;

/*dummy set to one if start year cohabitare is invalid*/
array fixyc1 (t) fixyc11 fixyc12;
array fixyc2 (t) fixyc21 fixyc22;
array fixyc3 (t) fixyc31 fixyc32;
   array fixyc (p) fixyc1 fixyc2 fixyc3;

/*dummy set to one if start year marry is invalid*/
array fixym1 (t) fixym11 fixym12;
array fixym2 (t) fixym21 fixym22;
array fixym3 (t) fixym31 fixym32;
   array fixym (p) fixym1 fixym2 fixym3;

array cems (p) cems1 cemsb cemsc;

if M54011r2>-4 then prevcmst=M54011r2;

```

```

else if m54011r2=-5 and m54011r1 gt -4 then
    prevcmst=m54011r1;
if M54012r2>-4 then M45002r2=M54012r2;
else if m54012r2=-5 and m54012r1 gt -4 then
m45002r2=m54012r1;

do p=1 to 3;
do t=1 to 2;
if cdli=1 then ymsc=dliym;
if mdli=1 then do;
    cmars=1; ymsm=dliym; end;

*****Legally separated at dli;
if mdli=3 then do;
    cmars=2; ymss=dliym;
end;
***if mdli is don't know or refuse then marital status is
same;
if -2 le mdli le -1 then cmars=mdli;
*if year and month start cohab is valid, then make
ymsc;
if ysc>0 and msc>0 then ymsc=(ysc*100)+msc;
**set dummy vars for invalid m/y start cohab;
if -3 le ysc le 0 then fixyc=1;
if -3 le msc le 0 then do; fixmc=1; end;
if ysc>0 and msc>0 then ymsc=(ysc*100)+msc;
if -3 le ysc le 0 or -3 le msc le 0 then ymsc=dliym;

/*need to add stop information for arrays*/
if yec>0 and mec>0 then ymec=(yec*100)+mec;
**Create Y/M end cohabiting;
**if year/month end cohabiting is missing and not
cohabiting at dli and not cohabiting continuously with
new partner, then ymec=int date;
if (-3 le yec le 0 or -3 le mec le 0) and cbeg ne 1 and
    cdli ne 1 and cbeg2 ne 1 then do;
    ymec=iym; end;

**Create vars for start and end of cohabitation in
    continuous months;
csmc=(round(ymsc,100)-198000)*.12+(ymsc-
    round(ymsc,100));
cemc=(round(ymec,100)-198000)*.12+(ymec-
    round(ymec,100));

**If cohab sdli or married at beginning of cohab,
    then set cohab end date as int date;
if cbeg=1 or cdli=1 or cbeg2=1 then cemc=doicm;

**If married when begin cohab, then assume
    married and change date arrays and fix flags;
if mbeg=1 then do;
    cmars=1; ymsm=ymsc; ymsc=.;
    if fixmc=1 then fixmm=1;
    if fixyc=1 then fixym=1;

end;

**If marital status changes since dli record what the
change was and how that affected start/end
dates
for marriage and cohab;

if how1=1 and hym1>0 and hmm1>0 then do;
    cmars=1; ymsm=(hym1*100)+hmm1; end;
if -2 le how1 le -1 then cmars=how1;

if how2=1 and hym2>0 and hmm2>0 then do;
    cmars=1; ymsm=(hym2*100)+hmm2; end;
if how2=3 and hys2>0 and hms2>0 then do;
    cmars=2; ymss=(hys2*100)+hms2; end;
if how2=4 and hys2>0 and hms2>0 then do;
    cmars=3; ymss=(hys2*100)+hms2; end;
if how2=5 and hys2>0 and hms2>0 then do;
    cmars=0; ymss=(hys2*100)+hms2; end;
if -2 le how2 le -1 then cmars=how2;

if how3=1 and hym3>0 and hmm3>0 then do;
    cmars=1; ymsm=(hym3*100)+hmm3; end;
if how3=3 and hys3>0 and hms3>0 then do;
    cmars=2; ymss=(hys3*100)+hms3; end;
if -2 le how3 le -1 then cmars=how3;

if how4=1 and hym4>0 and hmm4>0 then do;
    cmars=1; ymsm=(hym4*100)+hmm4; end;
if how4=3 and hys4>0 and hms4>0 then do;
    cmars=2; ymss=(hys4*100)+hms4; end;
if -2 le how4 le -1 then cmars=how4;

csmm=(round(ymsm,100)-198000)*.12+(ymsm-
    round(ymsm,100));
csms=(round(ymss,100)-198000)*.12+(ymss-
    round(ymss,100));
if csms>0 then cems=doicm;
if csmm>0 then cemm=doicm;

if csmc>0 then do;
    ttlc=ttlc+1; nuc=ttlc;
    if nuc=1 then cohcm=csmc;
end;

if csmm>0 then do;
    ttlm=ttlm+1; num=ttlm;
    if num=1 then marcm=csmm;
end;

corrc=0; corrm=0;
if ttlc>-1 or ttlm>-1 then do;
    if csmca2>0 and csmca2 ne . then do;
        corrc=corrc+1; end;
    if csmcb2>0 and csmcb2 ne . then do;
        corrc=corrc+1; end;

```

```

if csmcc2>0 and csmcc2 ne . then do;
    corrc=corrc+1; end;
if csmma2>0 and csmma2 ne . then do;
    corrm=corrm+1; end;
if csmb2>0 and csmb2 ne . then do;
    corrm=corrm+1; end;
if csmmc2>0 and csmmc2 ne . then do;
    corrm=corrm+1; end;
end;

ttlnew=ttlc-corrc; ttlmnew=ttlm-corm;

probflag=0; C=0;
do L=1 to 244;
    C=C+1;
    if csmc>0 and cemc>0 and csmc LE C LE cemc
        then coh=1;
    if cmars=1 and csmm>0 and cemm>0 and csmm le
        c le cemm then m=1;
    if 2 le cmars le 3 and csms>0 and cems>0 and
        csms le c le cems then m=cmars;
    if -2 le cmars le -1 then m=cmars;
    if m ge 1 and coh ge 1 then probflag=1;
    if c=doicm then do;
        if -2 le m le -1 then mars=m;
        if m=0 and coh=1 then mars=1;
        if m=0 and coh=0 then mars=2;
        if m=1 and coh=1 then mars=3;
        if m=1 and coh=0 then mars=4;
        if m=2 and coh=1 then mars=5;
        if m=2 and coh=0 then mars=6;
        if m=3 and coh=1 then mars=7;
        if m=3 and coh=0 then mars=8;
        if m=4 and coh=1 then mars=9;
        if m=4 and coh=0 then mars=10;
        marstat=mars;
    end;
    end;
    end;
    end;

if 1 le marstat le 2 then cmarstat=0;
if 3 le marstat le 4 then cmarstat=1;
if 5 le marstat le 6 then cmarstat=2;
if 7 le marstat le 8 then cmarstat=3;
if 9 le marstat le 10 then cmarstat=4;
if -2 le marstat le -1 then cmarstat=marstat;

array cvcm cohcm marm;
array cvy cohy mary;
array cvm cohcm marm;

do over cvcm;
    if 241 le cvcm le 252 then do;
        cvy=2000; cvm=cvcm-240; end;
    if 229 le cvcm le 240 then do;
        cvy=1999; cvm=cvcm-228; end;
    if 217 le cvcm le 228 then do;
        cvy=1998; cvm=cvcm-216; end;
    if 205 le cvcm le 216 then do;
        cvy=1997; cvm=cvcm-204; end;
    if 193 le cvcm le 204 then do;
        cvy=1996; cvm=cvcm-192; end;
    if 181 le cvcm le 192 then do;
        cvy=1995; cvm=cvcm-180; end;
    if 169 le cvcm le 180 then do;
        cvy=1994; cvm=cvcm-168; end;
    if cvcm=. then do;
        cvcm=-4; cvy=-4; cvm=-4; end;
end;

/*correct for those who don't know the start month for
cohabiting*/
if fixmc11=1 and (m31001m<0 and m31001m>-4) then
do;
    cohy=m31001y; cohcm=-3; cohm=-3; end;
/*correct for those who don't know the start year for
cohabiting*/
if fixyc11=1 and (m31001y<0 and m31001y>-4) then
do;
    cohm=m31001m; cohcm=-3; cohy=-3; end;
/*correct for those who don't know either the start year
or month for cohabiting*/
if fixyc11=1 and fixmc11=1 then do;
    cohm=-3; cohcm=-3; cohy=-3; end;

/*correct for those who don't know start month for
marriage*/
if fixmm11=1 and (m31001y<0 and m31001y>-4) then
do;
    mary=m31001y; marcm=-3; marm=-3; end;
/*correct for those who don't know start year for
marriage*/
if fixym11=1 and (m31001m<0 and m31001m>-4) then
do;
    marm=m31001m; marcm=-3; mary=-3; end;
/*to correct for those who don't know either the start
year or month for marriage*/
if fixym11=1 and fixmm11=1 then do;
    marm=-3; marcm=-3; mary=-3; end;

/*if old date present and don't deny then use old date*/
if prevcohcm ne -4 and (m712=1 or m714=1 or (m712=0
and m718=1)) then do;
    cohcm=prvcohcm; cohy=prevcohy;
    cohm=prevcohcm;
    end;
    if prevmarm gt -4 and (m712=1 or m714=1) then do;
        marcm=prvmarcm; mary=prevmary;
        marm=prevmarm;
        end;

```

```

/*correct total marriages & cohabs to reflect previous*/
/*previously in relationship, none since interview*/
if (m712 eq 1 and m740 eq 0) or (ttlcnew=0 and m740
eq 0 and coh_ttl ne 0) then do;
    ttlcnew=coh_ttl;
    if mar_ttl ne 0 then ttlmnew=mar_ttl;
end;

/*previously in relationship, in more since interview*/
if m712=1 and m740=1 then do;
    ttlmnew=ttlmnew+mar_ttl;
    ttlcnew=ttlcnew+coh_ttl;
end;
/*in relationship at int date, no more since interview*/
if m714=1 and m730=0 then do;
    if mar_ttl ne 0 then ttlmnew=mar_ttl;
    ttlcnew=coh_ttl;
end;
/*in relationship at int date, in more since interview*/
if m714=1 and m730=1 then do;
    ttlmnew=ttlmnew+mar_ttl;
    ttlcnew=ttlcnew+coh_ttl;
end;

***Change cohab dates to previous and subtract one
from cohab total if r. never cohabited but received
cohab date questions in error;
if m740 eq 0 and ttlcnew ne coh_ttl then do;
    ttlcnew=coh_ttl;                      cohy=prevcohy;
    cohm=prevcohm;                      cohcm=prvcohcm;
end;

***Change marstat if r. said s/he was married and said
there was a spouse in hh, but misunderstood later
questions about changes in marital status;

fixflag=0;
if cmarstat eq 0 and m620=1 and spo_hous = 1 and
    pubid ne 8545 then do;
    cmarstat=1;
    if marstat=1 then marstat=3;
    if marstat=2 then marstat=4;
    fixflag=1;

```

```

ttlmnew=-3;
if mary in (-4,-5) then mary=-3;
if marm in (-4,-5) then marm=-3;
if marcm in (-4,-5) then marcm=-3;
end;

*hand edits for separation without marriage;
if pubid=7716 or pubid=8778 then do;
    marstat=1; cmarstat=0; end;

* hand edit for person who is separated but did not
know date and seemed to have a new marriage in
addition to marriage at dli but this is unclear;
if pubid=5775 then do;
    ttlmnew=-3; marstat=6; cmarstat=2; end;

/*hand edits for those with start date prior to 1994*/
if cohcm gt 0 and cohcm lt 120 then do;
    cohcm=-3; cohm=-3; cohy=-3; end;
if marcm gt 0 and marcm lt 120 then do;
    marcm=-3; marm=-3; mary=-3; end;

if pubid=7819 then do;
    cohm=8; cohy=1993;
end;

array mar_var ttlmnew ttlcnew;
do i=1 to 2;
    if marstat gt -4 and mar_var(i)=-4 then mar_var(i)=0;
end;

if m700=-4 then do;
    marstat=-4;      cmarstat=-4;      ttlmnew=-4;
    ttlcnew=-4;      cohcm=-4;      marcm=-4;
    cohy=-4;      mary=-4;      cohm=-4;
    marm=-4;
end;

if m700=-5 then do;
    marstat=-5;      cmarstat=-5;      ttlmnew=-5;
    ttlcnew=-5;      cohcm=-5;      marcm=-5;
    cohy=-5;      mary=-5;      cohm=-5;
    marm=-5;
end;

```

YOUTH'S FERTILITY AND CHILD STATUS

Variables Created:	CV_CHILD_BIRTH_DATE.xx_M CV_CHILD_DEATH_DATE.xx_M CV_CHILD_BIRTH_MONTH.xx CV_CHILD_STATUS.xx CV_BIO_CHILD_HH	CV_CHILD_BIRTH_DATE.xx_Y CV_CHILD_DEATH_DATE.xx_Y CV_CHILD_BIRTH_MONTH.xx CV_BIO_CHILD_NR
---------------------------	--	--

Variables Used

Name in Program	Question Name on CD	Name in Program	Question Name on CD
Round 1: pubid	PUBID	Round 3: f400 f56001m f56002m f59001, f59002 f6000m, y bdaym1, bdayy1 bdaym2, bdayy2 bdaym3, bdayy3 dead9901-dead9903 res1-res3 bid01-bid03 uid01-uid03	YFER-400 YFER-5600.01~M, ~Y YFER-5600.02~M, ~Y YFER-5900.01, .02 YFER-6000.01~M, ~Y BIOCHILD_BDATE.01-M, ~Y BIOCHILD_BDATE.02-M, ~Y BIOCHILD_BDATE.03-M, ~Y BIOCHILD_DEAD.01 BIOCHILD_RESIDE.01-03 BIOCHILD_ID.01-03 BIOCHILD_UID.01-03
Round 2: dead9801-dead9803 cvbirm1, cvbiry1 cvbirm2, cvbiry2 cvbirm3, cvbiry3 cvdeam1 cvdeay1 cvstat1-cvstat3	BIOCHILD_DEAD.01-03 CV_CHILD_BIRTH_DATE.01~M, ~Y CV_CHILD_BIRTH_DATE.02~M, ~Y CV_CHILD_BIRTH_DATE.03~M, ~Y CV_CHILD_DEATH_DATE.01~M, ~Y CV_CHILD_DEATH_DATE.01~Y CV_CHILD_STATUS.01-03		

Codes for Created Variables

Date of birth and death variables

Date variables are presented as both the actual month and year and the month number in a continuous month scheme.

Status variables

- 1 Adopted
- 2 Deceased
- 3 Non-resident, foster care
- 4 Non-resident, not adopted or in foster care
- 5 Resident

This program creates a number of variables describing the youth's fertility and the current status of the youth's children. For more information on the continuous month system, see appendix 7 in this document.

```
/* create hand-edit roster variables for 8 cases */
/* two bio-children in hh, not deceased */
if pubid=594 then do; res1=1; end;
/* one bio-child in hh, not deceased */
if pubid =1205 then do; res1=1; end;
/* one bio-child, deceased when he was born */
if pubid =4177 then do;
    bdaym1=11; bdayy1=1998; bid1=201; res1=0;
end;
/* two bio-children in hh, both not deceased */
if pubid =6763 then do; res2=1; end;
/* first baby deceased, death date is missing; second
   baby in hh, not deceased; third baby in hh, not deceased
*/
if pubid =8729 then do;
    bid1=101; bdaym1=11; bdayy1=1992; res1=0;
    bid2=102; bdaym2=2; bdayy2=1996; res2=1;
    bid3=201; bdaym3=9; bdayy3=1998; res3=1;
end;

/* first, create a variable indicating dobm(i), birth
month, and doby(i), birth year, for each biological
child.*/
array bdaym(3) bdaym1-bdaym3;
array bdayy(3) bdayy1-bdayy3;
array dobm(3) dobm1-dobm3;
array doby(3) doby1-doby3;

do i=1 to 3;
    dobm(i)=-4;
    if bdaym(i) eq -3 then dobm(i)=3;
    if bdaym(i) gt 0 then dobm(i)=bdaym(i);
end;

do i=1 to 3;
    doby(i)=-4;
    if bdayy(i) eq -3 then doby(i)=-3;
    if bdayy(i) gt 0 then doby(i)=bdayy(i);
end;
```

```

do i=1 to 3;
  if bdaym(i)=-5 then do; dobm(i)=-5; doby(i)=-5; end;
end;

/* second, create a continuous month scheme variable
for the month of birth of the children using the formula:
(12*(doby(i)-1980)+dobm(i)) */
array mob(3) mob1-mob3;

do i=1 to 3;
  mob(i)=-4;
  if dobm(i) eq -3 then mob(i)=-3;
  if dobm(i) gt 0 then mob(i)=12*(doby(i)-
    1980)+dobm(i);
  if dobm(i) eq -5 then mob(i)=-5;
end;

/* third, create an actual date variable for the date of
death of the youth's children */
array dodm[3] dodm1-dodm3;
array dody[3] dody1-dody3;

do i=1 to 3; dodm(i)=-4; dody(i)=-4; end;

/* Due to an oversight of the questionnaire design (R1
through R3), the dates of death were not collected for
all the applicable cases. Here the program picked up
information for some cases from YFER-4800-Loop,
and assigned invalid skip (-3) to the rest. The error in
the fertility questionnaire has been correct in R4. */

/* The following 8 cases include all deceased bio
children reported from R1 to R3. */

if pubid =2426 then do; dodm1=8; dody1=1999; end;
if pubid =2936 then do; dodm1=1; dody1=1999; end;
if pubid =4177 then do; dodm1=11; dody1=1998; end;
if pubid =8405 then do; dodm1=11; dody1=1998; end;
if pubid =8729 then do; dodm1=-3; dody1=-3; end;

do i=1 to 3;
  if bdaym(i)=-5 then do;
    dodm(i)=-5; dody(i)=-5; end;
end;

/* fourth, create a continuous month scheme variable
for the month of death of the children using the
formula: (12*(dody(i)-1980)+dodm(i)) */
array mod(3) mod1-mod3;

do i=1 to 3;
  if dodm(i)=-4 then mod(i)=-4;
  if dodm(i)=-3 then mod(i)=-3;
  if dodm(i) gt 0 and dody(i) ge 1980 then
    mod(i)=12*(dody(i)-1980)+dodm(i);
  if dodm(i)=-5 then mod(i)=-5;
end;

  end;

/* fifth, create a variable indicating the status of youth's
first (second, third) child */
array res[3] res1-res3;
array status[3] status1-status3 ;
array flag[3] flag1-flag3;

/* Due to a similar questionnaire design error as
mentioned above, the status of biochild not living in the
household was not updated in R2 and R3.
Fortunately, the questionnaire collected this information
for some cases from YFER-4800-Loop. The following
program assigned invalid skip (-3) to the other cases
with biochild not living in the household */

do i=1 to 3; status(i)=-4;

/* Biochild not living in the household. The status was
not updated for some cases. YFER-4800-Loop
variables were used to determine the status for some
cases */

if res(i)=0 then do;
  status(i)=-3;

  /* adjusted by f59001 variable*/
  if bdayy(i) = f56001y and bdaym(i) = f56001m then
    do;
      if f59001 gt 0 then status(i)=4;
      if f59001=1 then status(i)=5;
      if f59001=3 then status(i)=1;
      if f59001=4 then status(i)=3;
    end;

  /* adjusted by f59002 variable*/
  if bdayy(i) = f56002y and bdaym(i) = f56002m then
    do;
      if f59002 gt 0 then status(i)=4;
      if f59002=1 then status(i)=5;
      if f59002=3 then status(i)=1;
      if f59002=4 then status(i)=3;
    end;
  end;

  **** if biochildren are deceased ****
  if (dodm(i) ne -4 and dodm(i) ne -5) then status(i)=2;

  **** biochild living in the household ****
  if res(i) eq 1 then status(i)=5;
  flag(i)=0;
  if res(i)=0 and f56001m gt 0 then flag(i)=1;
  if mob1=-5 then status(i)=-5;
end;

```

```

/* sixth, the number of children ever born and residing
in the household (tbiores) */
array biores[3] biores1-biores3;

/*initialize the biores variable and create tbiores */
do i=1 to 3;
  biores(i)=0; if status(i) eq 5 then biores(i)=1;
end;

tbiores=biores1+biores2+biores3;
if (mob1=-4 and mob2=-4 and mob3=-4) then
  tbionres=-4;
if mob1=-5 then tbionres=-5;

/* seventh, the number of children ever born and not
residing in the household (tbionres) */
array bionres[3] bionres1-bionres3;

do i=1 to 3;
  bionres(i)=0;
  if (status(i) eq 1 or status(i) eq 3 or status(i) eq 4 or
      res(i)=0) and status(i) ne 2 then bionres(i)=1;
end;

tbionres=bionres1+bionres2+bionres3;
if (mob1=-4 and mob2=-4 and mob3=-4) then
  tbionres=-4;
if mob1=-5 then tbionres=-5;

/* sort created variables by birthdays, so that the first
child listed is the oldest child. */
rmob1=0;      rmob2=0;      rmob3=0;
array rmob[3] rmob1 rmob2 rmob3;
array m[3] m1 m2 m3;

do i=1 to 3; rmob(i)=mob(i); end;

do i=1 to 3; if mob(i)=-3 then rmob(i)=1000; end;

m1=-4; m2=-4; m3=-4;

/* consider the families with one child */
if (mob1>-4 and mob2=-4 and mob3=-4) or (mob1=-4
and mob2>-4 and mob3=-4) or (mob1=-4 and mob2=-4
and mob3>-4) then do;
  if mob1>-4 then m1=mob1;
  if mob2>-4 then m1=mob2;
  if mob3>-4 then m1=mob3;
end;

/* consider families with 2 children */
if (mob1>-4 and mob2>-4 and mob3=-4) then do;
  if mob1 ge mob2 then do; m1=mob2; m2=mob1; end;
  if mob1<mob2 then do; m1=mob1; m2=mob2; end;
end;

```

```

if (mob1=-4 and mob2>-4 and mob3>-4) then do;
  if mob2 ge mob3 then do; m1=mob3; m2=mob2; end;
  if mob2<mob3 then do; m1=mob2; m2=mob3; end;
end;
if (mob1>-4 and mob2=-4 and mob3>-4) then do;
  if mob1 ge mob3 then do; m1=mob3; m2=mob1; end;
  if mob1<mob3 then do; m1=mob1; m2=mob3; end;
end;

/* consider families with three children */
if (mob1>-4 and mob2>-4 and mob3>-4) then do;
  m1=min(mob1, mob2, mob3);
  m3=max(mob1, mob2, mob3);
end;
do i=1 to 3;
  if mob(i) ne m1 and mob(i) ne m3 then m2=mob(i);
end;

array cv99bm[3] cv99bm1-cv99bm3;
array cv99by[3] cv99by1-cv99by3;
array cv99mob[3] cv99mob1-cv99mob3;
array cv99dm[3] cv99dm1-cv99dm3;
array cv99dy[3] cv99dy1-cv99dy3;
array cv99mod[3] cv99mod1-cv99mod3;
array cv99stat[3] cv99sta1-cv99sta3;

do i=1 to 3;
  if m1=mob(i) then do;
    cv99bm1=dobm(i);      cv99by1=doby(i);
    cv99mob1=rmob(i);     cv99dm1=dodm(i);
    cv99dy1=dody(i);     cv99mod1=mod(i);
    cv99sta1=status(i);
  end;
end;

do i=1 to 3;
  if m2=mob(i) then do;
    cv99bm2=dobm(i);      cv99by2=doby(i);
    cv99mob2=rmob(i);     cv99dm2=dodm(i);
    cv99dy2=dody(i);     cv99mod2=mod(i);
    cv99sta2=status(i);
  end;
end;

do i=1 to 3;
  if m3=mob(i) then do;
    cv99bm3=dobm(i);      cv99by3=doby(i);
    cv99mob3=rmob(i);     cv99dm3=dodm(i);
    cv99dy3=dody(i);     cv99mod3=mod(i);
    cv99sta3=status(i);
  end;
end;

/* non-interview cases*/
if f400=-5 then do;
  do i=1 to 3;

```

Appendix 3: Family Background and Formation Variable Creation

```
cv99bm[i]=-5; cv99by[i]=-5;    cv99mob[i]=-5;  
cv99dm[i]=-5; cv99dy[i]=-5;    cv99mod[i]=-5;  
cv99stat[i]=-5;  
end;                                end;  
endsas;
```

NUMBER OF RESIDENCES SINCE AGE 12**Variables Created:** CV_TTL_RESIDENCES**Variables Used**

Name in Program	Question Name on CD
CVr2	CV_TTL_RESIDENCES (round 2)
Y3500	YHHI-3500
Y3600	YHHI-3600

This program calculates the number of residences in which youth has lived since age 12. In round 1, the variable was created with information from the parent interview. In round 2, information collected from the respondent was combined with the round 1 variable to update the previous information. Subsequent rounds continue to update the variable with respondent-provided data.

```
/* Initialize each case to -4 */      RESID=-4;

***** respondents interviewed in round 2 *****
if cvR2=-1 or y3600=-1 then resid=-1;      /* refusals in either previous rounds or round 3 */
if cvR2=-2 or y3600=-2 then resid=-2;      /* don't know in either previous rounds or round 3 */
if cvR2=-3 then resid=-3;                  /* parents not interviewed in round 1 */

/* Non-negative answers in all three rounds, then add up the numbers of residences */
if cvR2 ge 0 and y3600 ge 0 then resid=cvR2+y3600;
if cvR2 ge 0 and y3500=0 then resid=cvR2;

***** respondents not interviewed in round 2 *****
if cvR2=-5 then do;
  if cvR1=-1 then resid=-1;          /* refusals in round 1 */
  if cvR1=-2 then resid=-2;          /* don't know in round 1 */
  if cvR1=-4 then resid=-3;          /* parents not interviewed in round 1 */

/* Non-negative answers in round 1, then add up cvR1 and the number of different residences since the round 1
interview */
  if cvR1 ge 0 and y3500=0 then resid=cvR1;
  if cvR1 ge 0 and y3600 ge 0 then resid=cvR1+y3600;

end;

***** respondents not interviewed in round 3 *****
if y3500=-5 then resid=-5;

endsas;
```